Fingerprint spoofing detection 2023

Machine Learning and Pattern Recognition

August 2023

Contents

1. Introduction

The goal of the project is the development of a classifier to detect whether a fingerprint image represents an authentic fingerprint or a spoofed fingerprint, i.e. a fingerprint image obtained by cloning, possibly with different methods, the fingerprint of an individual. The fingerprint images are represented by means of embeddings, i.e. low-dimensional representations of images obtained by mapping images to a low-dimensional manifold (typically few hundred dimensions).

2. Dataset information

To keep the model tractable, the dataset consists of synthetic data, and embeddings have significantly lower dimension than in real use-cases. The embeddings are 10-dimensional, continuous-valued vectors, belonging to either the authentic fingerprint (label 1) or the spoofed fingerprint (label 0) class. The embedding components do not have a physical interpretation.

The training dataset contains 2325 samples divided into 1525 spoofed fingerprints (class 0) and 800 authentic fingerprint (class 1), so it is very biased towards spoofed fingerprints.

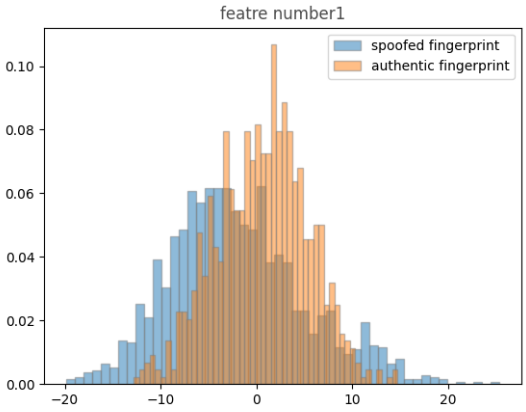
The test dataset contains 7704 samples divided into 5304 spoofed fingerprints (class 0) and 2400 authentic fingerprint (class 1).

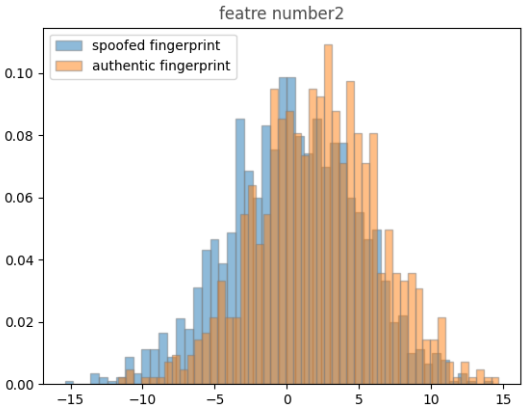
3. Dataset Analysis

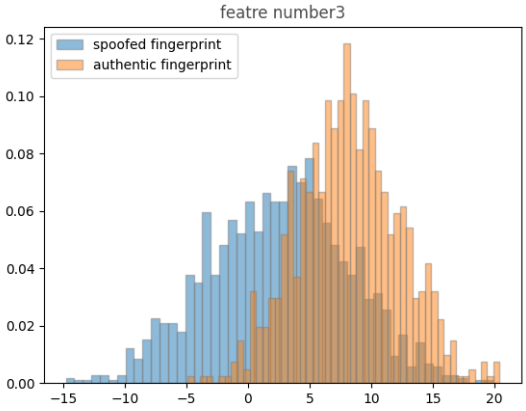
Dataset analysis is a crucial step in understanding and deriving insights from data. It involves systematically examining the characteristics, patterns, and relationships within a dataset.

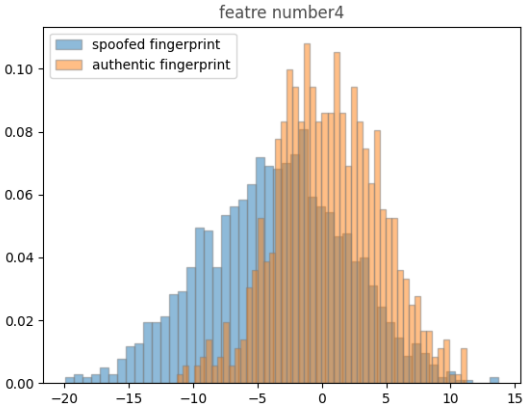
3.1. Feature analysis

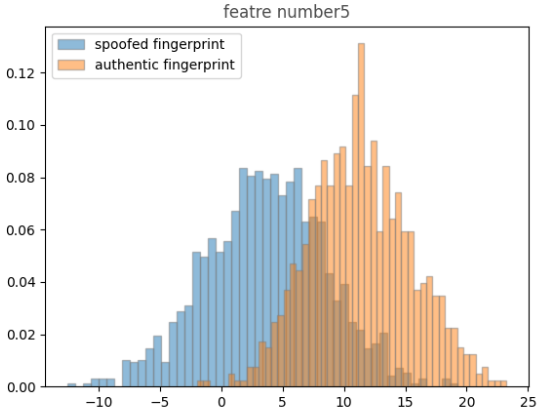
First, we draw the features we have using a histogram. a histogram is a graphical representation of the distribution of a dataset. It displays the frequency or count of data points falling into various ranges or "bins". Here we set the bins to 50.

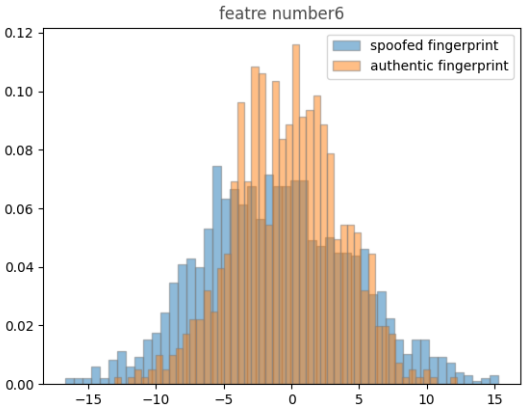

featre number1
spoofed fingerprint
authentic fingerprint

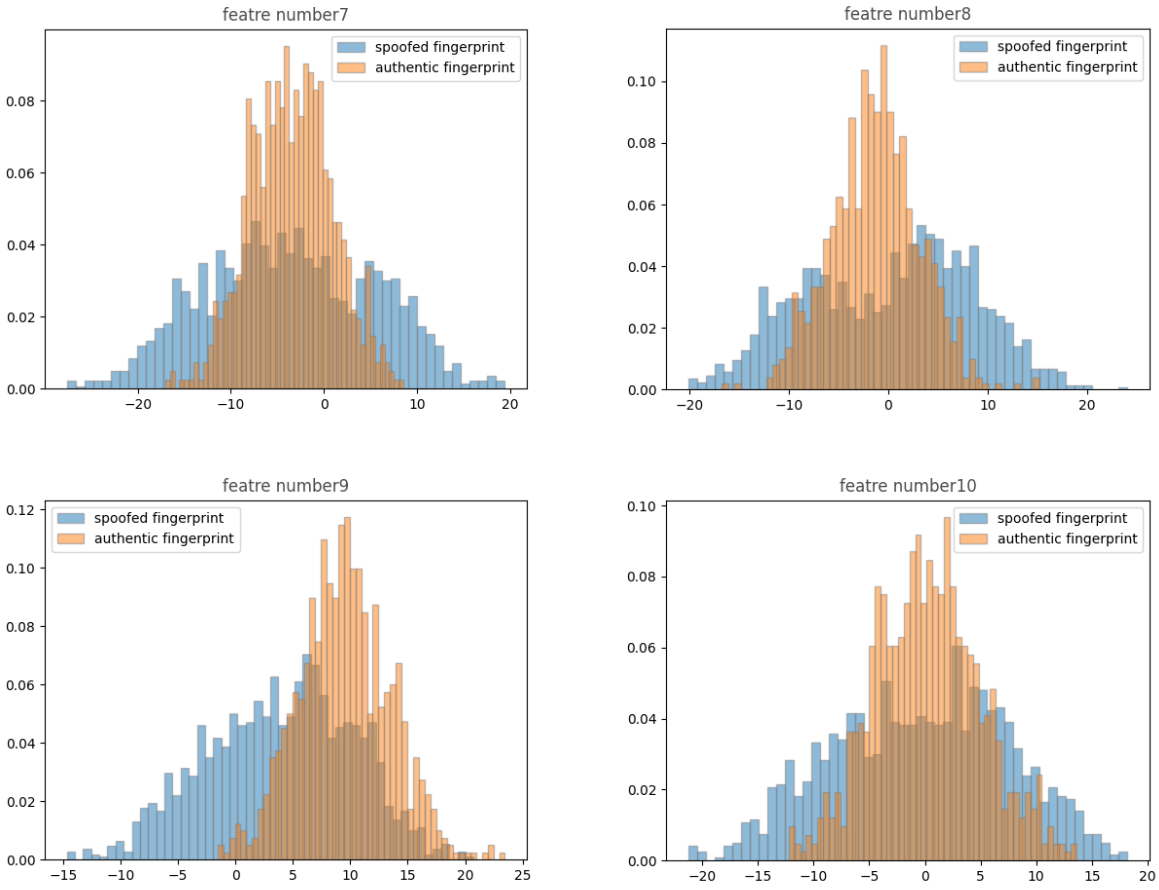

featre number2
spoofed fingerprint
authentic fingerprint


featre number3
spoofed fingerprint
authentic fingerprint


featre number4
spoofed fingerprint
authentic fingerprint


featre number5
spoofed fingerprint
authentic fingerprint

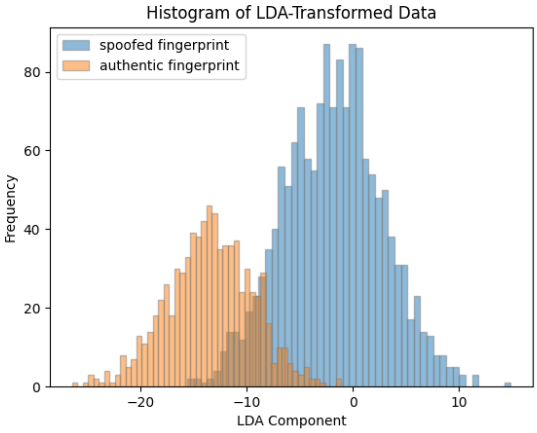

featre number6
spoofed fingerprint
authentic fingerprint



As can be seen, the authentic fingerprint class follows a Gaussian distribution which can be clearly seen in features 1, 3 and 8, but this is not the case in the spoof class.

3.2. Linear Discriminant Analysis (LDA)

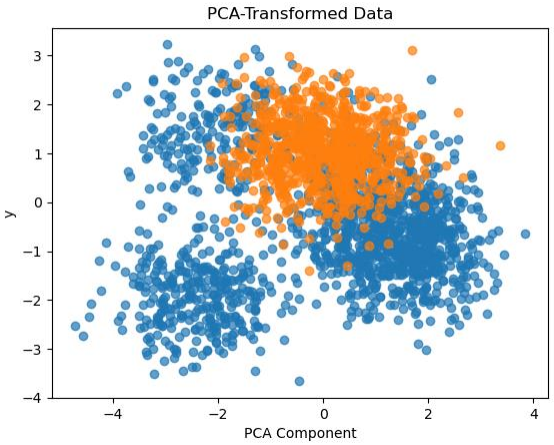
Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique used in statistics and machine learning. It's primarily used for finding a linear combination of features that maximizes the separation between different classes in a dataset. LDA is often applied in the context of classification problems, where the goal is to assign data points to predefined classes.



3.3. Principal Component Analysis (PCA)

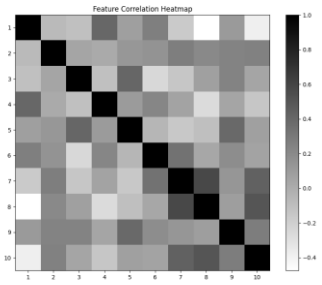
PCA stands for Principal Component Analysis. It's a statistical technique used for dimensionality reduction and data compression while preserving as much of the original variability as possible. PCA is widely used in various fields such as machine learning, image processing, and data analysis.

The main goal of PCA is to transform the original data, which might be high-dimensional, into a new coordinate system (a new set of features) where the features are uncorrelated. These new features are called "principal components." The first principal component captures the most variance in the data, the second principal component captures the second most, and so on. By reducing the dimensionality of the data, PCA can make subsequent analysis more efficient and reduce the risk of overfitting in machine learning models. We used pca2 for better understand and display.

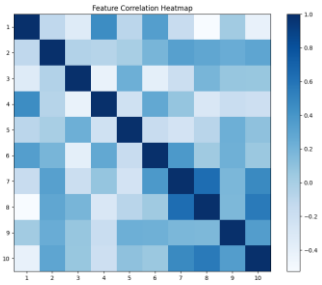


3.4. correlation heatmaps

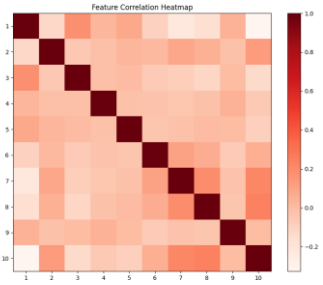
In this section we want to understand the correlation between features.
A correlation heatmap is a graphical representation that uses colors to visualize the correlation between features in a dataset. Correlation measures the statistical relationship between two variables, indicating whether and how they tend to change together. A correlation heatmap displays these correlation coefficients in a matrix form, with each cell representing the correlation between a pair of variables.



All data



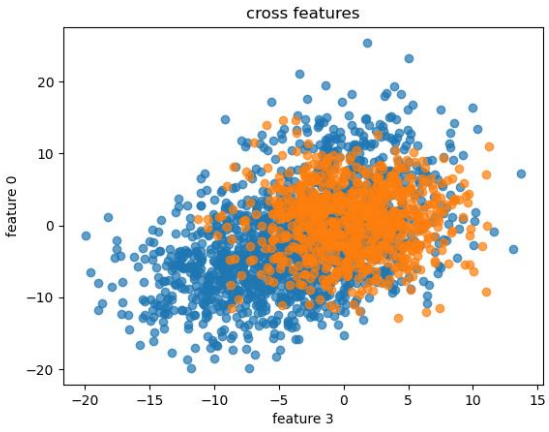
spoofed fingerprints

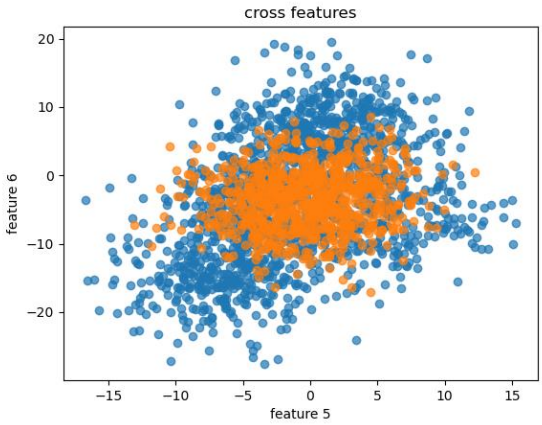


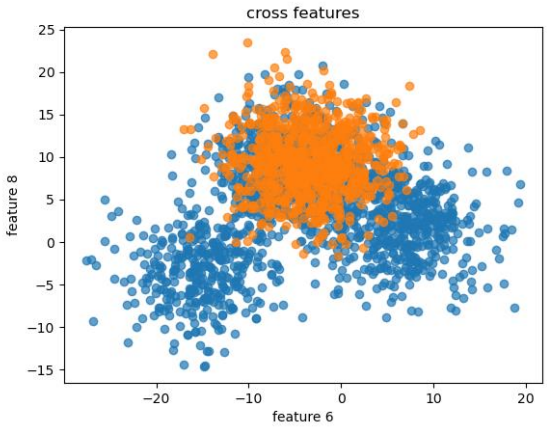
authentic fingerprint

3.5. Cross features plot

Cross-features plots are graphical representations used to visualize the relationships between multiple pairs of features in a dataset. These plots are especially useful when you want to understand how your features interact with each other. In a cross-features plot, each variable is represented on one axis of the plot, and the data points are plotted as dots on the graph. Each dot corresponds to a data entry and its position on the plot is determined by the values of the two variables it represents.







4. Validation Methodology

We use several models for classifiers. Also we use K-fold and PCA too.
K-fold cross-validation is a technique used in machine learning and model evaluation to assess the performance of a model in a more robust and reliable manner. It involves dividing the dataset into "k" subsets or folds of approximately equal size. The model is then trained and evaluated "k" times, using different combinations of folds for training and testing in each iteration. In this project, we used k=5 and different values of 9, 8, 7 for PCA.
It's important to note that our dataset is highly unbalanced, meaning that the number of instances in different classes is significantly imbalanced.
The target application considers an application where prior probabilities of authentic and spoofed classes are the same, but labeling a spoofed fingerprint as authentic has a larger cost due to the security vulnerabilities that such errors would create. The target application working point is therefore defined by the triplet (π_T = 0.5, C_{fn} = 1, C_{fp} = 10).

4.1. Gaussian Models

Gaussian Models work under the assumption that data follows a normal distribution. In this section we compare performances of 3 different models:

- Multivariate Gaussian Classifier
- Naive Bayes Gaussian Classifier
- MVG with Tied Covariances

According to the plots and histograms that we drew in the previous step, we understood that the data follows the Gaussian distribution, so we can predict that the Multivariate Gaussian (MVG) model can perform well.
Our analysis of the histograms further exposes dissimilar data distributions across individual classes, leading to the emergence of discernibly separate covariance matrices. This circumstance leads us to the notion that the performance of MVG classifier with Tied Covariance might fall short of the performance exhibited by other classifiers.

PCA	Z-Norm	MVG	MVG + Tied	Naive Bayes
None	False	**0.325**	0.476	0.468
9	False	0.344	0.476	0.371
8	False	0.332	0.472	0.379
7	False	0.329	0.462	0.381
None	True	**0.325**	0.476	0.468
9	True	0.344	0.476	0.371
8	True	0.332	0.472	0.379
7	True	0.329	0.462	0.381

minDCF for Gaussian models

As expected, the MVG model performs best. It can also be seen that z-norm does not have much effect on performance.

4.2. Discriminative Models

Next, our attention turns to discriminative models. In this section we compare performances of 2 different models:

- Logistic Regression Model (LR)
- Quadratic Logistic Regression Model (QLR)

These models are designed to directly estimate the posterior distribution using the sigmoid function. our objective will be:

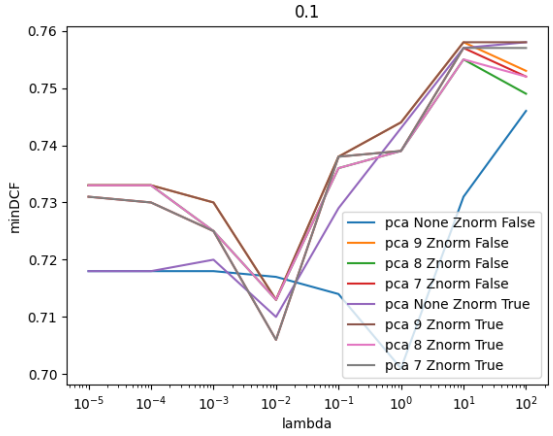
$$R(w) = \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{\pi_T}{n_T}\sum_{i|z_i=1} l(z_i s_i) + \frac{1-\pi_T}{n_F}\sum_{i|z_i=-1} l(z_i s_i)$$

In the formula above, si is equal to:

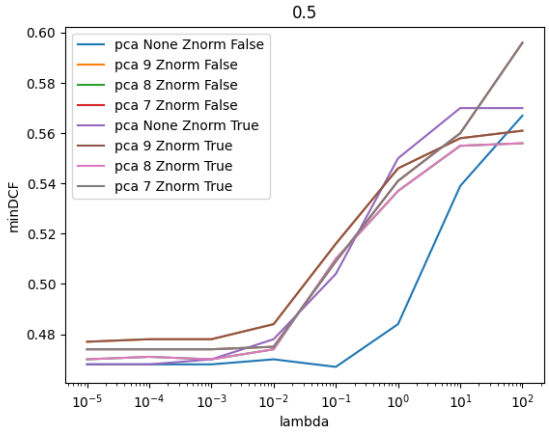
$$\boldsymbol{w^T x + b}$$

In the above formulas **w** and **b** are the model parameters and $\frac{\lambda}{2}||\boldsymbol{w}||^2$ is a regularization term that helps obtaining a **w** with lower norm.
Since the data distribution was Gaussian and the performance of the MVG model was good on it, q is expected to perform better.
Quadratic Logistic Regression (QLR) is a variant of the traditional Logistic Regression (LR) model that incorporates quadratic terms in addition to the linear terms used in standard logistic regression. In standard logistic regression, the model estimates the log-odds of a binary outcome as a linear combination of input features. This log-odds value is then transformed using the sigmoid function to obtain the probability of the positive class. Quadratic Logistic Regression extends this concept by allowing for quadratic relationships between input features and the log-odds of the outcome. This means that the model can capture more complex interactions and patterns in the data that may not be adequately represented by a purely linear relationship. We use:

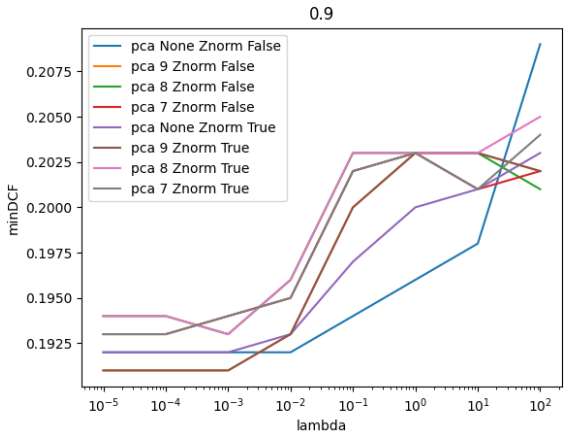
$$\phi(x) = \begin{pmatrix} \text{vec}(xx^T) \\ x \end{pmatrix}$$



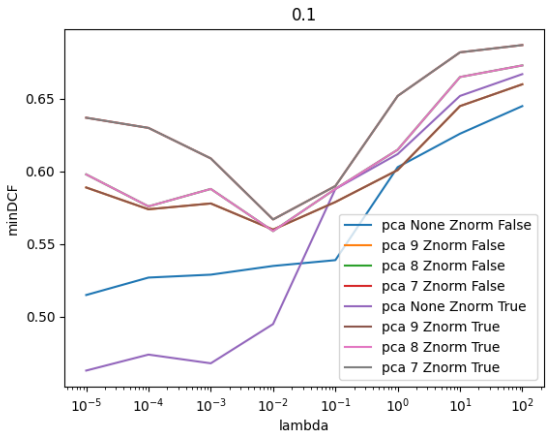
LR with $\pi_T = 0.1$



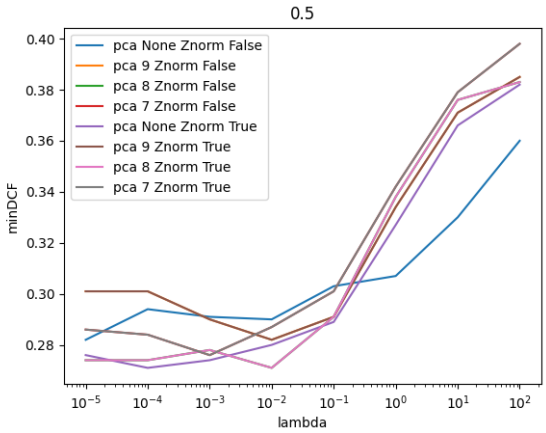
LR with $\pi_T = 0.5$



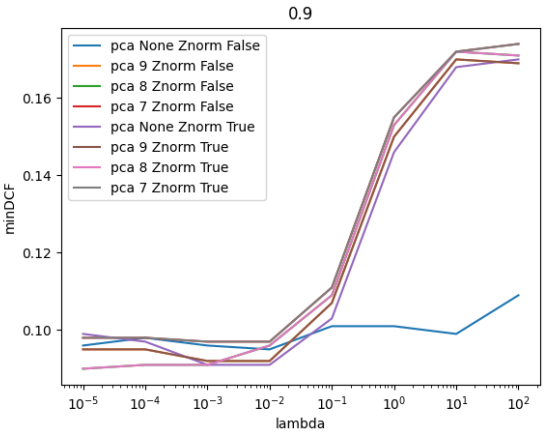
LR with $\pi_T = 0.9$



QLR with $\pi_T = 0.1$



QLR with $\pi_T = 0.5$



QLR with $\pi_T = 0.9$

We tested the graphs with three values of pi. We also used three values for PCA with z-norm and without z-norm. As we predicted, it has performed better in all quadratic situations.

Now let's see the results in the form of numbers and tables:

LR with πT = 0.1:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.718	0.718	0.718	0.717	0.714	**0.701**	0.731	0.746
9	False	0.733	0.733	0.730	0.713	0.738	0.744	0.758	0.753
8	False	0.733	0.733	0.725	0.713	0.736	0.739	0.755	0.749
7	False	0.731	0.730	0.725	0.706	0.738	0.739	0.757	0.752
None	True	0.718	0.718	0.720	0.710	0.729	0.743	0.757	0.758
9	True	0.733	0.733	0.730	0.713	0.738	0.744	0.758	0.758
8	True	0.733	0.733	0.725	0.713	0.736	0.739	0.755	0.752
7	True	0.731	0.730	0.725	0.706	0.738	0.739	0.757	0.757

LR with πT = 0.5:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.468	0.468	0.468	0.470	**0.467**	0.484	0.539	0.567
9	False	0.477	0.478	0.478	0.484	0.516	0.546	0.558	0.561
8	False	0.470	0.471	0.470	0.474	0.510	0.537	0.555	0.556
7	False	0.474	0.474	0.474	0.475	0.509	0.541	0.560	0.596
None	True	0.468	0.468	0.470	0.478	0.504	0.550	0.570	0.570
9	True	0.477	0.478	0.478	0.484	0.516	0.546	0.558	0.561
8	True	0.470	0.471	0.470	0.474	0.510	0.537	0.555	0.556
7	True	0.474	0.474	0.474	0.475	0.509	0.541	0.560	0.596

LR with πT = 0.9:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.192	0.192	0.192	0.192	0.194	0.196	0.198	0.209
9	False	**0.191**	**0.191**	**0.191**	0.193	0.200	0.203	0.203	0.202
8	False	0.194	0.194	0.193	0.196	0.203	0.203	0.203	0.201

7	False	0.193	0.193	0.194	0.195	0.202	0.203	0.201	0.202
None	True	0.192	0.192	0.192	0.193	0.197	0.200	0.201	0.203
9	True	**0.191**	**0.191**	**0.191**	0.193	0.200	0.203	0.203	0.202
8	True	0.194	0.194	0.193	0.196	0.203	0.203	0.203	0.205
7	True	0.193	0.193	0.194	0.195	0.202	0.203	0.201	0.204

QLR with $\pi T = 0.1$:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.515	0.527	0.529	0.535	0.539	0.603	0.626	0.645
9	False	0.589	0.574	0.578	0.560	0.579	0.601	0.645	0.660
8	False	0.598	0.576	0.588	0.559	0.588	0.615	0.665	0.673
7	False	0.637	0.630	0.609	0.567	0.590	0.652	0.682	0.687
None	True	**0.463**	0.474	0.468	0.495	0.588	0.612	0.652	0.667
9	True	0.589	0.574	0.578	0.560	0.579	0.601	0.645	0.660
8	True	0.598	0.576	0.588	0.559	0.588	0.615	0.665	0.673
7	True	0.637	0.630	0.609	0.567	0.590	0.652	0.682	0.687

QLR with $\pi T = 0.5$:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.282	0.294	0.291	0.290	0.303	0.307	0.330	0.360
9	False	0.301	0.301	0.290	0.282	0.291	0.334	0.371	0.385
8	False	0.274	0.274	0.278	**0.271**	0.291	0.338	0.376	0.383
7	False	0.286	0.284	0.276	0.287	0.301	0.342	0.379	0.398
None	True	0.276	**0.271**	0.274	0.280	0.289	0.327	0.366	0.382
9	True	0.301	0.301	0.290	0.282	0.291	0.334	0.371	0.385
8	True	0.274	0.274	0.278	**0.271**	0.291	0.338	0.376	0.383
7	True	0.286	0.284	0.276	0.287	0.301	0.342	0.379	0.398

QLR with $\pi T = 0.9$:

PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.096	0.098	0.096	0.095	0.101	0.101	0.099	0.109
9	False	0.095	0.095	0.092	0.092	0.107	0.150	0.170	0.169
8	False	**0.090**	0.091	0.091	0.096	0.109	0.153	0.172	0.171
7	False	0.098	0.098	0.097	0.097	0.111	0.155	0.172	0.174
None	True	0.099	0.097	0.091	0.091	0.103	0.146	0.168	0.170
9	True	0.095	0.095	0.092	0.092	0.107	0.150	0.170	0.169
8	True	**0.090**	0.091	0.091	0.096	0.109	0.153	0.172	0.171
7	True	0.098	0.098	0.097	0.097	0.111	0.155	0.172	0.174

The choice of λ is crucial and requires tuning. A small value of λ might not have a significant effect on the model, while a large value can lead to too much simplification, causing underfitting.
As it can be seen, znorm has no effect on improving the performance of the model when PCA is used.

4.3. No Probabilistic Models

Now we consider the no probabilistic models like Support Vector Machine models (SVMs). we compare performances of 2 different models:

- Linear
- Polynomial kernel
- Radial Basis Function kernel (RBF)

we will consider the prior-weighted version of the model (dual problem SVM).
The dual problem of the prior-weighted Support Vector Machine (SVM) involves transforming the primal optimization problem (minimizing a function subject to constraints) into a different form that can be solved more efficiently. The dual problem involves Lagrange multipliers or dual variables associated with the constraints of the primal problem.
For the prior-weighted SVM, the primal problem is given by minimize:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\omega_i\xi_i$$

subject to:

$$y_i(w.x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

Where the dual problem is formulated as follows maximize:

$$\arg\max_{\alpha} \boldsymbol{\alpha}^T\mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha}$$

$$\text{s.t. } \sum_{i=1}^{n}\alpha_i z_i = 0\,, \quad 0 \leq \alpha_i \leq C_i,\ i = 1,\ldots,n$$

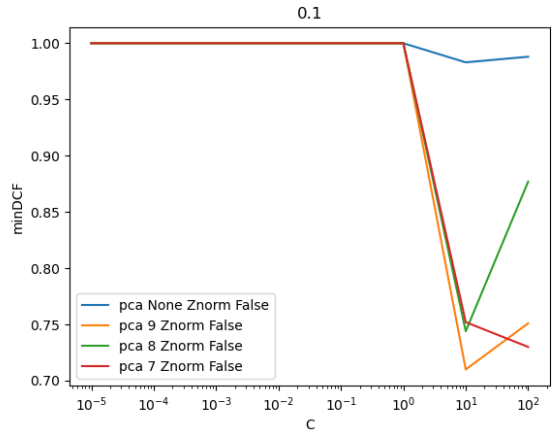
Also we select:

$$C_T = C\frac{\pi_T}{\pi_T^{emp}} \ \ and \ \ C_F = C\frac{\pi_F}{\pi_F^{emp}}$$

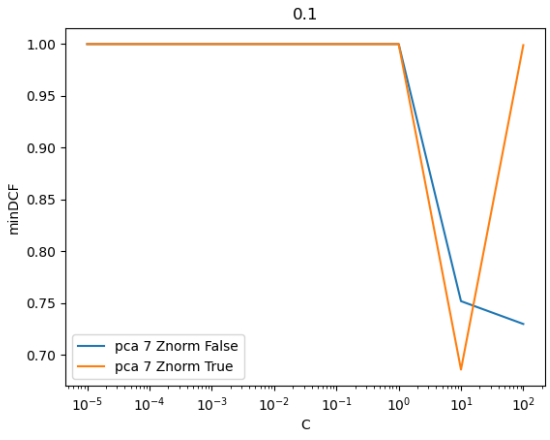
π_T^{emp} and π_F^{emp} are the empirical priors (i.e. sample proportions) for the two classes computed over the training dataset.

As you can see, we have a hyper-parameter C that we set its value in the range of 10^{-5} to 10^{2}.

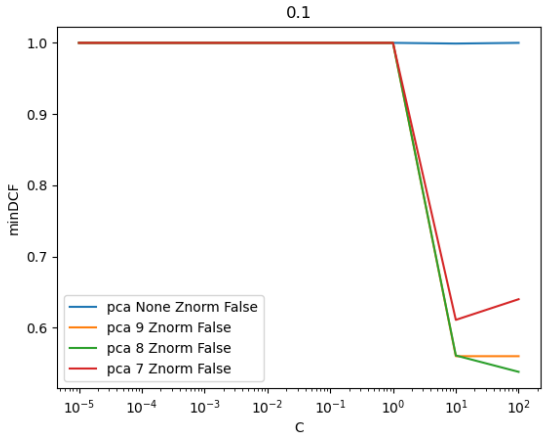
note that in all SVM we set K=1, in the SVM with Radial Basis Function kernel we set γ=0.1 and in the SVM with Polynomial kernel we set d=2.



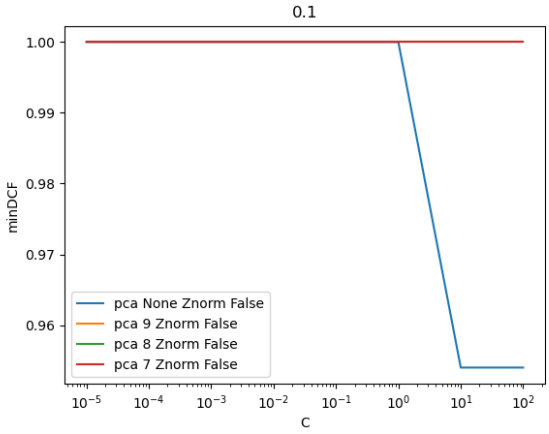
Linear SVM with Z-norm False



compare Linear SVM between PCA 7 with and without Z-norm



SVM with polynomial kernel



SVM with RBF kernel

Now let's see the results in the form of numbers and tables:

Linear SVM:

PCA	ZNrom	$c=10^{-5}$	$c=10^{-4}$	$c=10^{-3}$	$c=10^{-2}$	$c=10^{-1}$	$c=10^{-0}$	$c=10^{1}$	$c=10^{2}$
None	False	1	1	1	1	1	1	0.983	0.988
9	False	1	1	1	1	1	1	**0.710**	0.751
8	False	1	1	1	1	1	1	0.744	0.877
7	False	1	1	1	1	1	1	0.752	0.730

Polynomial SVM:

PCA	ZNrom	$c=10^{-5}$	$c=10^{-4}$	$c=10^{-3}$	$c=10^{-2}$	$c=10^{-1}$	$c=10^{-0}$	$c=10^{1}$	$c=10^{2}$
None	False	1	1	1	1	1	1	0.999	1
9	False	1	1	1	1	1	1	0.56	0.56
8	False	1	1	1	1	1	1	0.561	**0.538**
7	False	1	1	1	1	1	1	0.611	0.640

RBF SVM:

PCA	ZNrom	$c=10^{-5}$	$c=10^{-4}$	$c=10^{-3}$	$c=10^{-2}$	$c=10^{-1}$	$c=10^{-0}$	$c=10^{1}$	$c=10^{2}$
None	False	1	1	1	1	1	1	**0.952**	**0.952**

9	False	1	1	1	1	1	1	1	1
8	False	1	1	1	1	1	1	1	1
7	False	1	1	1	1	1	1	1	1

4.4 Gaussian Mixture Model – GMM

The last type of classifier that we will consider is the Gaussian Mixture Model. In general, this type of model tends to be used for density estimation problem and the assumption is that our data can be distributed as Gaussian with one or more components.

PCA	target	nontarget	Z-norm	minDCF
None	0	2	False	0.258
9	0	2	False	0.269
8	0	2	False	0.261
7	0	2	False	0.262
None	0	2	True	0.258
9	0	2	True	0.269
8	0	2	True	0.261
7	0	2	True	0.262
None	0	3	False	0.244
9	0	3	False	0.284
8	0	3	False	0.280
7	0	3	False	0.276
None	0	3	True	0.244
9	0	3	True	0.284
8	0	3	True	0.280
7	0	3	True	0.276
None	0	4	False	0.241
9	0	4	False	0.282
8	0	4	False	0.268
7	0	4	False	0.269
None	0	4	True	0.235
9	0	4	True	0.282
8	0	4	True	0.268
7	0	4	True	0.269
None	1	2	False	0.258

9	1	2	False	0.269
8	1	2	False	0.261
7	1	2	False	0.262
None	1	2	True	0.258
9	1	2	True	0.269
8	1	2	True	0.261
7	1	2	True	0.262
None	1	3	False	0.244
9	1	3	False	0.284
8	1	3	False	0.280
7	1	3	False	0.276
None	1	3	True	0.244
9	1	3	True	0.284
8	1	3	True	0.280
7	1	3	True	0.276
None	1	4	False	0.241
9	1	4	False	0.282
8	1	4	False	0.268
7	1	4	False	0.269
None	1	4	True	0.235
9	1	4	True	0.282
8	1	4	True	0.268
7	1	4	True	0.269

5. Evaluation

In this section we want to evaluate the models on the test data.
At this stage, we only review the models that performed better. For example, in SVM Models, we only check polynomial.

5.1. Gaussian Models

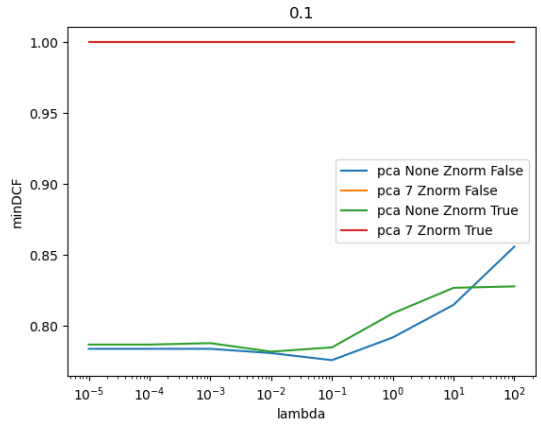
First, we examine the Gaussian models that had an acceptable performance in validation:

PCA	Z-Norm	MVG	MVG + Tied	Naive Bayes
None	False	**0.276**	0.455	0.352
9	False	0.258	0.469	0.323

5.2. Discriminative Models

We use all the λ we used in the training here. It should also be noted that we set the value of pi to 0.1. Let's see the results:

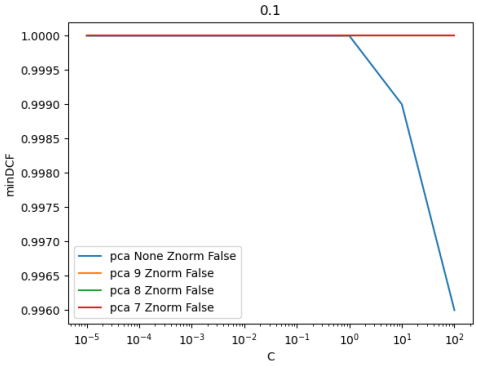
PCA	ZNrom	$\lambda=10^{-5}$	$\lambda=10^{-4}$	$\lambda=10^{-3}$	$\lambda=10^{-2}$	$\lambda=10^{-1}$	$\lambda=10^{-0}$	$\lambda=10^{1}$	$\lambda=10^{2}$
None	False	0.784	0.784	0.784	0.781	0.776	0.792	0.815	0.856
7	False	1	1	1	1	1	1	1	1
None	True	0.787	0.787	0.788	0.782	0.785	0.809	0.827	0.884
7	True	1	1	1	1	1	1	1	1



5.3. No Probabilistic Models

In this section, we will only examine the polynomial SVM. As with validation, we also set the degree value to 2. The results on the test data will be as follows:

PCA	ZNrom	$c=10^{-5}$	$c=10^{-4}$	$c=10^{-3}$	$c=10^{-2}$	$c=10^{-1}$	$c=10^{-0}$	$c=10^{1}$	$c=10^{2}$
None	False	1	1	1	1	1	1	0.999	0.996
9	False	1	1	1	1	1	1	1	1
8	False	1	1	1	1	1	1	1	1
7	False	1	1	1	1	1	1	1	1



5.4. Gaussian Mixture Model – GMM

Finally, we consider the Gaussian Mixture Models:

PCA	target	nontarget	Z-norm	minDCF
None	0	2	False	0.229
None	0	2	True	0.274
None	0	3	False	**0.219**
None	0	3	True	0.266
None	1	2	False	0.229
None	1	2	True	0.274
None	1	3	False	**0.219**
None	1	3	True	0.266

6. Conclusion

Our conclusion from the obtained results shows that there is a great similarity between validation and evaluation data. Also, according to the data distribution and as we predicted, GMM and MVG models showed the best performance.